

## **IN THE CLAIMS:**

1. (Original) A method for generating a configurator comprising:  
creating a customizable product, the customizable product including a set of one or more attributes to define the customizable product.  
assigning the customizable product to a customizable product class, the customizable product class is a parent class of a hierarchy defining the configurator;  
adding a component product class to the customizable product class, the component product class is a subclass of the customizable product;  
adding a customizable class rule to the customizable product class, the customizable class rule including expressions which provide conditions on component products added to the customizable product; and  
mapping a customizable UI to the customizable product class, the customizable UI to provide access structure to the configurator.
2. (Original) The method of claim 1 wherein the component product class includes component product subclasses.
3. (Original) The method of claim 1 wherein the component product class inherits the attributes of the customizable product class.
4. (Original) The method of claim 1 further comprising:  
adding one or more component product classes to a port; and  
adding the port to the customizable product class, the port to allow the configurator to classify a group of component products.
5. (Original) The method of claim 4 wherein the port includes a cardinality attribute, the cardinality attribute to constrain the number of component products to be added by the configurator.

6. (Original) The method of claim 4 wherein the cardinality attribute includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain the minimum number of component products to be added by the configurator, the maximum cardinality to constrain the maximum number of component products to be added by the configurator.
7. (Original) The method of claim 1 wherein the cardinality attribute includes a default cardinality, the default cardinality defines a quantity of the component product class to be added by the configurator.
8. (Original) The method of claim 1 wherein the class rules includes a collection of expressions, the expressions include a property path, a constant, and a operator.
9. (Original) The method of claim 1 wherein the class rule includes a natural language syntax.
10. (Original) The method of claim 1 wherein the class rule is a subclass of the customizable product class.
11. (Withdrawn) The method of claim 1 wherein the mapping includes building the customizable UI from a set of themes, groups, and controls.
12. (Withdrawn) The method of claim 11 wherein the themes are tabs and wizards.
13. (Withdrawn) The method of claim 11 wherein each theme in the set of themes, groups, and controls includes at least one of the set of background colors, fonts, and multi-lingual.
14. (Withdrawn) The method of claim 11 wherein the group includes one or more of the controls.
15. (Withdrawn) The method of claim 11 wherein the control includes one or more of a drop down box, a radio button, and a list box.
16. (Withdrawn) The method of claim 1 wherein the customizable UI is used to generate a user interface for a component product class.
17. (Withdrawn) The method of claim 1 wherein the customizable UI is a subclass of the customizable product.
18. (Withdrawn) The method of claim 1 wherein the customizable UI is used to generate a configurator user interface with HTML, Applets, and ActiveX programming languages.

19. (Original) The method of claim 1 wherein the component product class includes a static attribute, the static attribute is not associated with a parent class.
20. (Original) The method of claim 1 wherein the component product class, customizable class rules, and UI class are object oriented classes.
21. (Original) The method of claim 1 wherein the customizable product comprises a object oriented structure.
22. (Original) The method of claim 1 wherein the customizable product includes versioning.
23. (Original) The method of claim 1 wherein the configurator is stored in a data store.
24. (Original) A machine-readable medium that provides instructions, which when executed by a set of one or more processors, cause the set of processors to perform operations for generating a configurator comprising:
  - creating a customizable product, the customizable product including a set of one or more attributes to define the customizable product.
  - assigning the customizable product to a customizable product class, the customizable product class is the parent class of a hierarchy defining the configurator;
  - adding a component product class to the customizable product class, the component product class is a subclass of the customizable product;
  - adding a customizable class rule to the customizable product class, the customizable class rule including expressions which provide conditions on component products added to the customizable product; and
  - mapping a customizable UI to the customizable product class, the customizable UI to provide access structure to the configurator.
25. (Original) The machine-readable medium of claim 24 wherein the component product class includes component product subclasses.
26. (Original) The machine-readable medium of claim 24 wherein the component product class inherits the attributes of the customizable product class.

27. (Original) The machine-readable medium of claim 24 further comprising:  
adding one or more component product classes to a port; and  
adding the port to the customizable product class, the port to allow the configurator to  
classify a group of component products.
28. (Original) The machine-readable medium of claim 27 wherein the port includes a cardinality attribute, the cardinality attribute to constrain the number of component products to be added by the configurator.
29. (Original) The machine-readable medium of claim 27 wherein the cardinality attribute includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain the minimum number of component products to be added by the configurator, the maximum cardinality to constrain the maximum number of component products to be added by the configurator.
30. (Original) The machine-readable medium of claim 24 wherein the cardinality attribute includes a default cardinality, the default cardinality defines a quantity of the component product class to be added by the configurator.
31. (Original) The machine-readable medium of claim 24 wherein the class rules includes a collection of expressions, the expressions include a property path, a constant, and an operator.
32. (Original) The machine-readable medium of claim 24 wherein the class rule includes a natural language syntax.
33. (Original) The machine-readable medium of claim 24 wherein the class rule is a subclass of the customizable product class.
34. (Withdrawn) The machine-readable medium of claim 24 wherein the mapping to include building the customizable UI from a set of themes, groups, and controls.
35. (Withdrawn) The machine-readable medium of claim 34 wherein the themes includes tabs and wizards.

36. (Withdrawn) The machine-readable medium of claim 34 wherein the theme includes background color, fonts, and multi-lingual.
37. (Withdrawn) The machine-readable medium of claim 34 wherein the group includes one or more of the controls.
38. (Withdrawn) The machine-readable medium of claim 34 wherein the control includes one or more of a drop down box, a radio button, and a list box.
39. (Withdrawn) The machine-readable medium of claim 24 wherein the customizable UI is used to generate a user interface for a component product class.
40. (Withdrawn) The machine-readable medium of claim 24 wherein the customizable UI is a subclass of the customizable product.
41. (Withdrawn) The machine-readable medium of claim 24 wherein the customizable UI is used to generate a configurator user interface with HTML, Applets, and ActiveX programming languages.
42. (Original) The machine-readable medium of claim 24 wherein the component product class includes a static attribute, the static attribute is not associated with a parent class.
43. (Original) The machine-readable medium of claim 24 wherein the component product class, customizable class rules, and UI class are object oriented classes.
44. (Original) The machine-readable medium of claim 24 wherein the customizable product comprising a object oriented structure.
45. (Original) The machine-readable medium of claim 24 wherein the customizable product includes versioning.
46. (Original) The machine-readable medium of claim 24 wherein the configurator is stored in a data store.
47. (Original) An object oriented configurator comprising:  
a customizable product class;

a component product, the component product is a subclass of the customizable product, the component product inherits a set of one or more attributes from the customizable product class;

a customizable class rule, the customizable class rule having a collection of expressions constrain the component product; and

a customizable UI, the customizable UI is mapped to the customizable product providing a view of the component product.

48. (Original) The object oriented configurator in claim 47 further comprising:  
a port, the port comprising a set of one or more of the component products.
49. (Original) The object oriented configurator in claim 48 wherein the port includes a cardinality, the cardinality to constrain a number of component products to add to the customizable product class.
50. (Original) The object oriented configurator in claim 49 wherein the cardinality includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain a minimum number of component products to be added by the configurator, the maximum cardinality to constrain a maximum number of component products to be added by the configurator.
51. (Original) The object oriented configurator in claim 49 wherein the cardinality includes a default cardinality, the default cardinality defines a quantity of the component product class to be added by the configurator.
52. (Original) The object oriented configurator in claim 47 wherein the customizable class rule, and customizable UI are subclasses of the customizable product.
53. (Original) The object oriented configurator in claim 47 wherein the component product includes a static attribute, the static attribute is not inherited from a parent class.

54. (Original) The object oriented configurator in claim 47 wherein the attribute is of type string, number, date, and Boolean.
55. (Original) The object oriented configurator in claim 47 further comprising a second customizable product.
56. (Original) The object oriented configurator in claim 55 wherein the component product includes one or more of a second customizable product.
57. (Original) The object oriented configurator in claim 47 wherein the customizable class rule includes expressions.
58. (Original) The object oriented configurator in claim 47 wherein the collection of expressions includes a constant, a property path, and operators.
59. (Original) The object oriented configurator in claim 47 wherein the collection of expressions use natural language syntax.
60. (Original) The object oriented configurator in claim 47 wherein the customizable class rule inherit from the customizable product class.
61. (Original) The object oriented configurator in claim 47 wherein the class rule is a subclass of the customizable product.
62. (Original) The object oriented configurator in claim 47 wherein the class rule includes a natural language template, the natural language template uses a natural language syntax.
63. (Original) The object oriented configurator in claim 47 wherein the class rule trigger actions to be performed by the object oriented configurator.
64. (Original) The object oriented configurator in claim 47 wherein the class rule is designed with a template.
65. (Original) The object oriented configurator in claim 47 wherein the class rule includes a batch rule and a predicate-based customizable class rule.
66. (Original) The object oriented configurator in claim 47 wherein the class rule is stored in a data store.

67. (Original) The object oriented configurator in claim 47 wherein the expression includes a property path.
68. (Original) The object oriented configurator in claim 47 further comprising:  
a operator property, the operator property enabling a function to be performed within the expression.
69. (Original) The object oriented configurator in claim 47 further comprising:  
a script, the script to communicate with another application.
70. (Withdrawn) The object oriented configurator in claim 47 wherein the customizable UI includes a theme, group, and control.
71. (Withdrawn) The object oriented configurator in claim 47 wherein the theme includes a tab, a wizard, a font, and a color.
72. (Withdrawn) The object oriented configurator in claim 47 wherein the control includes one or more of a drop down box, a radio button, and a list box.
73. (Withdrawn) The object oriented configurator in claim 47 wherein the customizable UI map is used to generate a configurator user interface with HTML, JAVA applets, and ActiveX components.
74. (Original) The object oriented configurator in claim 47 wherein each component product class has an unique identifier, the unique identifier is used to locate an associative component product.
75. (Original) The object oriented configurator in claim 47 further comprising link items.
76. (Original) A configurator modeling system comprising:  
a customizable product modeling component, the customizable product modeling component to build a customizable product model,  
a customizable class rule modeling component, the customizable class rule modeling component to provide conditions for adding the set of one or more component products classes to the configurator; and



- a UI structuring component, the UI structuring component to map a graphical user interface.
77. (Original) The configurator of claim 76 wherein the customizable product model includes:  
a set of one or more customizable product classes, the customizable product modeling  
component to define the set of one or more customizable product classes; and  
a set of one or more component product classes, the customizable product modeling  
component to define the set of one or more component product classes, the  
customizable product to define attributes for the set of one or more component  
products.
78. (Original) The configurator modeling system of claim 76 the system is designed by a  
configuration expert.
79. (Withdrawn) A customizable UI comprising:  
a UI theme, the UI theme including a set of properties, the UI theme selected from a set of  
theme templates;  
a UI control, the UI control associated with a set of customizable product attributes; and  
a UI group, the UI group includes a set of one or more of the UI controls.
80. (Withdrawn) The customizable UI of claim 79 wherein the customizable UI is mapped to the  
customizable product to provide a graphical user interface.
81. (Original) A machine-readable medium that provides instructions, which when executed by a  
set of one or more processors, cause the set of processors to perform operations comprising:  
creating a product structure, the product structure including a set of one or more configurable  
complex objects describing the product structure;  
assigning the product structure to a class;  
adding one or more component products to the product structure;  
adding a customizable rule; and  
mapping a UI, the UI to provide a graphical user interface to the product structure.

82. (Withdrawn) A method of building a customizable UI comprising:  
selecting a theme, the theme to display the background from which to display a customizable product;  
setting properties of the theme;  
adding a group to the theme, the group to include a set of one or more controls;  
associating the controls with a set of one or more attributes of the customizable products; and  
associating the controls to the group.
83. (Original) A configurator comprising:  
triggering configuring rules that determine which component products may be selected; and  
determining which component products to be selected to the configurator.
84. (Original) A customizable rule configurator comprising:  
a customizable model component, the customizable model component to generate a customizable product structure; and  
a rule builder component, the rule builder component to generate a customizable class rules, the customizable class rule to constraint entries to a customizable product structure.
85. (Original) The customizable rule configurator of claim 84 wherein the customizable model component includes a customizable product class, the customizable class rule is a subclass of the customizable product class.
86. (Original) The customizable rule configurator of claim 84 wherein the customizable class rule includes an expression.
87. (Original) The customizable rule configurator of claim 86 wherein the expression includes a constant, a property path, and an operator.
88. (Original) The customizable rule configurator of claim 86 wherein the expression includes natural language syntax.
89. (Original) The customizable rule configurator of claim 86 wherein the expression is provided by the configuration expert by a template.

90. (Original) The customizable rule configurator of claim 86 wherein the expression includes a operator property.
91. (Original) The customizable rule configurator of claim 89 wherein the template allows for a natural language syntax.
92. (Original) The customizable rule configurator of claim 84 wherein the rule builder component includes a batch customizable class rule template.
93. (Original) The customizable rule configurator of claim 84 wherein the rule builder component includes a predicate-based customizable class rule template.
94. (Original) The customizable rule configurator of claim 84 wherein the customizable class rule is stored in a data store.
95. (Withdrawn) An apparatus composed of logic blocks to customize a product comprising:  
a first logic block to create a customizable product, the customizable product including a set of one or more attributes to define the customizable product.  
a second logic block to assign the customizable product to a customizable product class, the customizable product class is a parent class of a hierarchy defining the configurator;  
a third logic block to add a component product class to the customizable product class, the component product class is a subclass of the customizable product;  
a fourth logic block to add a customizable class rule to the customizable product class, the customizable class rule including expressions which provide conditions on component products added to the customizable product; and  
a fifth logic block to map a customizable UI to the customizable product class, the customizable UI to provide access structure to the configurator.
96. (Withdrawn) The apparatus of claim 1 wherein the component product class includes component product subclasses.
97. (Withdrawn) The apparatus of claim 1 wherein the component product class inherits the attributes of the customizable product class.

98. (Withdrawn) The apparatus of claim 1 further comprising:  
a sixth logic block to add one or more component product classes to a port; and  
a seventh logic block to add the port to the customizable product class, the port to allow the configurator to classify a group of component products.
99. (Withdrawn) The apparatus of claim 4 wherein the port includes a cardinality attribute, the cardinality attribute to constrain the number of component products to be added by the configurator.
100. (Withdrawn) The apparatus of claim 4 wherein the cardinality attribute includes a minimum cardinality and a maximum cardinality, the minimum cardinality to constrain the minimum number of component products to be added by the configurator, the maximum cardinality to constrain the maximum number of component products to be added by the configurator.
101. (Withdrawn) The apparatus of claim 1 wherein the cardinality attribute includes a default cardinality, the default cardinality defines a quantity of the component product class to be added by the configurator.
102. (Withdrawn) The apparatus of claim 1 wherein the class rules includes a collection of expressions, the expressions include a property path, a constant, and a operator.
103. (Withdrawn) The apparatus of claim 1 wherein the class rule includes a natural language syntax.
104. (Withdrawn) The apparatus of claim 1 wherein the class rule is a subclass of the customizable product class.
105. (Withdrawn) The apparatus of claim 1 wherein the fifth logic block to map includes building the customizable UI from a set of themes, groups, and controls.
106. (Withdrawn) The apparatus of claim 11 wherein the themes are tabs and wizards.
107. (Withdrawn) The apparatus of claim 11 wherein each theme in the set of themes, groups, and controls includes at least one of the set of background colors, fonts, and multi-lingual.

108. (Withdrawn) The apparatus of claim 11 wherein the group includes one or more of the controls.
109. (Withdrawn) The apparatus of claim 11 wherein the control includes one or more of a drop down box, a radio button, and a list box.
110. (Withdrawn) The apparatus of claim 1 wherein the customizable UI is used to generate a user interface for a component product class.
111. (Withdrawn) The apparatus of claim 1 wherein the customizable UI is a subclass of the customizable product.
112. (Withdrawn) The apparatus of claim 1 wherein the customizable UI is used to generate a configurator user interface with HTML, Applets, and ActiveX programming languages.
113. (Withdrawn) The apparatus of claim 1 wherein the component product class includes a static attribute, the static attribute is not associated with a parent class.
114. (Withdrawn) The apparatus of claim 1 wherein the component product class, customizable class rules, and UI class are object oriented classes.
115. (Withdrawn) The apparatus of claim 1 wherein the customizable product comprises a object oriented structure.
116. (Withdrawn) The apparatus of claim 1 wherein the customizable product includes versioning.
117. (Withdrawn) The apparatus of claim 1 wherein the configurator is stored in a data store.